**Week 3 Seminar Solutions**

1. Give an Example of *soft fork*.

The classic example of a change that was made via soft fork is Pay-to-Script-

Hash (P2SH). P2SH was not present in the first version of the Bitcoin protocol. Its introduction caused a soft fork, because for old nodes, a valid P2SH transaction would still verify correctly. As interpreted by the old

nodes, the script is simple—it hashes one data value and checks whether the hash matches the value specified in the output script. Old nodes don’t carry out the (now required) additional step of running that value itself to see whether it is a valid script. We rely on new nodes to enforce the new rules (i.e., that the script

actually redeems this transaction).

P2SH was successful. It’s also possible that new cryptographic schemes could be added by a soft fork. We could also add some extra metadata in the **coinbase** parameter that has some agreed-on meaning. Today, any value is accepted in the coinbase parameter. But we could, in the future, require that the coinbase have some specific format. One proposed idea is that, in each new block, the coinbase includes the Merkle root of a tree containing the entire set of unspent transactions. It would only result in a soft fork, because old nodes might mine a block that didn’t have the required new coinbase parameter, so that block would be rejected by the network, but the old node would catch up and join the main chain that the network is mining.

2. Install the ‘bitcoin’ package in the Python and generate a random private key.

3. Find the public key from your private key.

4. Find the bitcoin address from your public key.

5. Generate a multisignature address with k=2, n=3, explain the meaning.

6. Find a valid address from Blockchain.com and check its history via Python.

2-6 See the Jupyter Notebook